L17: Entry 3 of 4 File: USPT Sep 15, 1998

DOCUMENT-IDENTIFIER: US 5809328 A
TITLE: Apparatus for fibre channel transmission having interface logic, buffer memory, multiplexor/control device, fibre channel controller, gigabit link module, microprocessor, and bus control device

Abstract Text (1):
The present invention is an apparatus for adapting transmissions between an industry standard data bus of a host computer having a host memory and a fiber channel coupled between said host computer and a peripheral storage subsystem having at least one disk drive, which apparatus comprises an interface logic coupled between the industry standard bus and a local bus of the apparatus; a buffer memory coupled to the local bus; a multiplexor/control device coupled to the local bus and being disposed for transmitting therethrough address and data; a fiber channel controller disposed for formatting header and data structures that meet fiber channel protocol, which controller is coupled to the multiplexor/control; a gigabit link module disposed for converting the header and data structures from a parallel format to a serial format and being coupled between the fiber channel controller and the fiber channel; a microprocessor disposed for providing service requests from the host to read and write data from the host memory to and from the peripheral storage subsystem via the buffer memory, the microprocessor is coupled to a processor bus; and, a bus control device coupled between the processor bus and the local bus for providing service requests of the interface logic, the fiber channel controller and the microprocessor, and for arbitrating control of the local bus.

Brief Summary Text (13):
The present invention is an apparatus for adapting transmissions between an industry standard data bus of a host computer having a host memory and a fibre channel coupled between said host computer and a peripheral storage subsystem having at least one disk drive, which apparatus comprises an interface logic coupled between the industry standard bus and a local bus of the apparatus; a buffer memory coupled to the local bus; a multiplexor/control device coupled to the local bus and being disposed for transmitting therethrough address and data; a fibre channel controller disposed for formatting header and data structures that meet fibre channel protocol, which controller is coupled to the multiplexor/control; a gigabit link module disposed for converting the header and data structures from a parallel format to a serial format and being coupled between the fibre channel controller and the fibre channel; a microprocessor disposed for providing service requests from the host to read and write data from the host memory to and from the peripheral storage subsystem via the buffer memory, the microprocessor is coupled to a processor bus; and, a bus control device coupled between the processor bus and the local bus for providing service requests of the interface logic, the fibre channel controller and the microprocessor, and for arbitrating control of the local bus.

Detailed Description Text (12):
The FC controller 31 manages the protocol for sending and receiving information on the fibre channel 14. To achieve this, the processor 22 and the FC controller 31 share data structures and data buffers that are maintained in the buffer memory 30. The FC controller 31 provides a connection between the GLM 32 and local bus 33/34 via the address/data bus 36 and the multiplexor/control 35.

Detailed Description Text (13):
The processor 22 can access the buffer memory 30 by using either addresses in processor region 9 or 11; wherein the memory configuration registers define region 9 as little endian and region 11 as big endian. The term "endian" refers to the sequence in which a multi-byte word is

transferred. For example, "little endian" means that the least significant byte is transferred first with the most significant byte being transferred last. The term "big endian" refers to the transfer of the most significant byte first, etc. This allows the processor to move information between two locations in the buffer memory (potentially the same location) and change the byte ordering at the same time. This ability to change endianess, by reading from one region and writing to another is useful when dealing with commands or status information. It is also useful when reading from or writing to disk drives that may be formatted differently.

Detailed Description Text (14):
A specific example of endianness occurs at the PCI interface logic 28. The PCI bus is in little endian format and the FC controller 31 assumes/supplies data in big endian format. In order to provide compatible data between the two systems, a byte swap is performed at the PCI interface 28 to the local bus 33/34. Thus, the PCI interface logic 28 provides the PCI bus 29 to local bus 33/34 interface. The logic 28 can function as either a bus master or target on both the PCI 29 and the local bus 33/34. In one embodiment a PCI 9060, which is available from PLX Technology, Inc. of Mountain View, Calif., was used for the logic 28. The PCI 9060 has four sets of internal registers and operates in three modes: The register access mode, the DMA mode and the Pass-Through mode.

Detailed Description Text (19):
At this juncture, it is determined if the command is a read or a write operation (decision diamond 52). For a write data operation, the processor 22 sets up shared data structures in the buffer memory 30 (block 53), which describes the operation and the data location for the FC Controller 31. Using the DMA controller within the PCI Interface Logic 28, the processor 22 moves the write data from the host memory 13 to the buffer memory 30 (block 54), and then informs the FC Controller 31 that a command and data are available by writing to registers within the FC Controller 31 (block 55). In preparation for the data transfer, the FC Controller 31 reads the shared data structures from the buffer memory 30, and then transfers data from the buffer to the GLM 32 (block 56), which performs a parallel to serial conversion (block 57). The data is subsequently transmitted over the Fibre Channel 14 to a disk drive for storage.

Detailed Description Text (20):
For a read data operation, a branch is taken from the diamond 52 to the flow chart illustrated in FIG. 3B as denoted by a connector "A". The processor 22 sets up shared structures in the buffer memory 30 that describe the operation and data destination for the FC controller 31 (block 58); and, then informs the FC Controller that a command is available by writing to registers therein (block 59). In preparation for the data transfer, the FC Controller 31 reads the shared data structures from the buffer memory 30 (block 60), and then transfers data from a disk drive over the fibre channel 14 via the GLM 32 (block 61) (which performs a serial to parallel conversion-block 62) to the buffer memory 30. Using the DMA controller within the PCI Interface Logic 28, the processor 22 directs moving of the read data from the buffer memory 30 to the host memory 13 (block 63).

CLAIMS:

1. An apparatus for adapting transmissions between an industry standard data bus of a host computer having a host memory and a fibre channel coupled between said host computer and a peripheral storage subsystem having at least one disk drive, said apparatus comprising:

a. an interface logic having a first input/output terminal coupled to said industry standard bus and a second input/output terminal coupled to a local bus of said apparatus, said interface logic comprising a first set of registers disposed for receiving requests from said host and a second set of registers disposed for receiving instructions from said microprocessor;

b. a buffer memory having input/output terminals coupled to said local bus;

c. a multiplexor/control device having a first set of input/output terminals coupled to said local bus and a second set of input/output terminals disposed for transmitting therethrough address and data, said multiplexor/control comprising bi-directional registers and buffers

disposed for merging address and data on said local bus before transmission to said fibre channel controller during a write to disk operation, and for separating address and data received from said gigalink module during a read from disk operation;

d. a fibre channel controller disposed for formatting header and data structures that meet fibre channel protocol, having a first input/output terminal coupled to said second set of input/output terminals of said multiplexor/control device and a second input/output terminal;

e. a gigabit link module disposed for converting said header and data structures from a parallel format to a serial format and having a first input/output terminal coupled to said second input/output terminal of said fibre channel controller and a second input/output terminal coupled to said fibre channel;

f. a microprocessor disposed for providing service requests from said host to read and write data from said host memory to and from said peripheral storage subsystem via said buffer memory, said microprocessor having address and data input/output terminals coupled to a processor bus; and,

g. a bus control device coupled between said processor bus and said local bus for providing service requests of said interface logic, said fibre channel controller and said microprocessor and to arbitrate control of said local bus.

5. In a file server having a peripheral storage subsystem coupled thereto, a method for adapting fibre channel transmissions between said file server and said storage subsystem to an industry standard data bus of said file server, said method comprising the steps of:

a. receiving a command control block from said file server, said block containing information as to type of command and information as to location in a memory of said file server of data to be transferred;

b. determining type of said command, and if a write command;

c. storing said data in a buffer memory;

d. moving said data from said buffer memory, through a fibre channel controller and to a gigabit link module;

e. converting in said gigabit link module said data from a parallel format to a serial format for transmission over a fibre channel; and,

f. formatting in said fibre channel controller said data into header and data structures that meet fibre channel protocol.

7. The method as in claim 6 further including the step of de-formatting said read data from header and data structures into address and data.

9. In a file server having a peripheral storage subsystem coupled thereto, a method for adapting fibre channel transmissions between said file server and said storage subsystem to an industry standard data bus of said file server, said method comprising the steps of:

a. receiving a command control block from said file server, said block containing information as to type of command and information as to location in a memory of said file server of data to be transferred;

b. determining type of said command, and if a write command;

c. storing said data in a buffer memory;

d. moving said data from said buffer memory, through a fibre channel controller and to a gigabit link module;

e. converting in said gigabit link module said data from a parallel format to a serial format for transmission over a fibre channel;

f. if said command is a read command, receiving from said storage subsystem via said fibre channel read data;

g. converting in said gigabit link module said read data from a serial format to a parallel format for receipt and transmission to said server;

h. moving said read data from said gigabit link module through said fibre channel controller to said buffer memory;

i. transferring said read data from said buffer memory to said memory in said file server; and,

j. formatting in said fibre channel controller said data into header and data structures that meet fibre channel protocol.

10. The method as in claim 9 further including the step of de-formatting said read data from header and data structures into address and data.

Previous Doc Next Doc Go to Doc#

First Hit Fwd Refs Previous Doc Next Doc Go to Doc#
End of Result Set

Generate Collection Print

L15: Entry 1 of 1 File: USPT May 23, 2000

DOCUMENT-IDENTIFIER: US 6067595 A
TITLE: Method and apparatus for enabling high-performance intelligent I/O subsystems using multi-port memories

Detailed Description Text (22):
The upper two bits from the look-up table define a 2-bit control field 116. The first bit of the control field 116 is used as a write protect bit and the second bit is used to control a byte swap engine of the PCI interface 68. This feature allows a page by page endianess conversion. The remaining fourteen bits from the lookup table form bits 16 through 29 of the PCI master address. The upper two PCI address bits are defined by a CSR register and define the PCI master base address. This grants a 1 GB PCI address window. The SCI-to-PCI LUT can be any memory size. If 512 address translation entries are not sufficient, a larger memory can be used, allowing a larger LUT index 112.

Detailed Description Text (29):
The simplest multi-port memories currently available are either dual-port or quad-port memories. It is possible to implement a triple-port memory using three dual-port memories. However, only one memory chip is required to build a triple-port memory 70, using a quad-port memory 130, as shown in FIG. 8. The quad-port memory has four 8-bit data buses. Therefore, four chips need to be combined to form a 32-bit bus. The SCI backend bus interface is typically 64 bits wide. In order to avoid using eight 8-bit multi-port memories to build a 64-bit bus, two data ports can be combined to form a 64-bit bus. The lowest address bit of the merged 64-bit bus is tied to a fixed level (low for port 2 and high for port 3). Which address bit is tied to which level depends on the IOP bus interface's endianess flavor and the organization of data bus B. If a 64-bit write transaction is executed, the quad-port memory recognizes two simultaneous write requests at two ports to two adjacent memory locations with identical timing. This demonstrates how the interface of the present invention can accommodate various bus widths, without requiring specially designed chips. All ports of the multi-port memory operate asynchronously allowing independent clock domains at each port. Another utilization of a quad-port memory is to use the fourth port as either an additional network port or allow addition of a second processor 74' if one processor 74 proves insufficient.

Detailed Description Text (30):
The configuration of a triple-port memory 70 using a quad-port memory 130 is shown in more detail in FIG. 9. The Blink address bus is 11 bits wide because the lowest address pin is tied to GND or VCC, respectively (A2(0)<=`0`; A3(0)<=`1` or vice versa, for all chips depending on the endianess of the involved buses). The PCI local bus chip B1 is asymmetric with the chip driving D8-D15, which contains part of the pass through address and part ATE address bits. Other groupings of the output enable signals and maybe the address buses are obviously possible depending on the specific fly-by address translation scheme.

Previous Doc Next Doc Go to Doc#

L17: Entry 2 of 4 File: USPT Dec 16, 2003

DOCUMENT-IDENTIFIER: US 6665746 B1
TITLE: System and method for prioritized context switching for streaming data memory transfers

Drawing Description Text (5):
FIG. 4 is a schematic diagram of a data structure for a scatter/gather list element according to one embodiment of the present invention.

Detailed Description Text (15):
With reference to FIG. 4, in one embodiment of the invention the data structure 400 for an element within the scatter/gather list typically includes a local address pointer 401, a system address pointer 402 (such as a PCI address pointer), a transfer configuration byte 404, a byte count 406, and a link address pointer 408.

Detailed Description Text (19):
Transfer configuration byte 404 includes a read/write bit (R/W bit) 450, an enable transfer interrupt bit (ENABLE XFER) 452, a link bit (LINK) 454, and an Endian byte swapping bit (ENDIAN BYTE SWAP) 456.

Detailed Description Text (23):
The Endian byte swapping bit 456 is an operation in which the data being transferred needs to be manipulated of its Endianess. Interfacing the different Endianess (little to big Endian or vise versa) is simplified if the scatter/gather element provides the information whether transfer of data needs to be Endianess manipulated or converted.